Community-wise Abstractive Text Summarization

Yiyang Li
Stanford University
yiyang7@stanford.edu

Yatong Chen
Stanford University
yatong@stanford.edu

Teng Zhang
Stanford University
tengz@stanford.edu

Abstract

Neural sequence-to-sequence models have provided a viable new approach for abstractive text summarization. However, most current state-of-art models are based on News dataset (like CNN/Daily Mail), the content of which always exhits a "summary-details" structure, resulting in an absence of multi-attention focus in the model. In this work, we consider the TL;DR Reddit dataset, which exhits a community-wise difference among contents, and it is captured by the information called "subreddit". In our work, we propose two frameworks based on the baseline pointer-generated model, aiming at capturing the community-wise variation between different subreddits. First, we perform domain encoding that incoporates the subreddit information into the embedding vectors before the input information goes into the encoder state. Second, we use Multi-Attention layer to train different subreddit differently, resulting in an overall improvement of the model's performance. We apply our model on the TL;DR Reddit dataset, outperforming the lead-2 and the result from the baseline model by at least 2 Rouge points for most of the subreddit.

1 Introduction

Text summarization is one of the prevailing tasks in modern NLP. Taking a text document as the input, the task aims at producing a document with shorter length preserving the main information in the input. Text summarization is also related to many other NLP tasks. Broadly speaking, it can be viewed as learning a map between sequences, thus sharing similar properties and research methodologies like machine translation, video captioning and question answering.

The two most common approaches to the task are extraction-based summarization, where the model extracts salient parts of the source document (we refer to as "content"), and abstraction-based summarization, where the model also concisely rephrase the salient information. Due to the huge success of deep learning models in NLP, more works are targeted at abstraction-based summarization for practical reasons while the two approaches share similar building blocks. In this paper, we will focus our work on the abstractive side.

Current abstractive summarization models are mostly built on news datasets, e.g. CNN/Daily Mail (Hermann et al., 2015) (the prevailing one), or more recently, Newsroom (Grusky et al., 2018). Despite being wildly utilized, news summary datasets have a major drawback, i.e. high uniformity in news content-summarization pairs. Professional journalists and reporters obey certain patterns when drafting the news stories, most of which are in the "summary-detail" structure. In other words, using the starting sentences as the summarization for news articles would generally be good enough, and this is verified repeatedly, as shown in Table 1.

Model	ROUGE-1	ROUGE-2	ROUGE-L
Best	41.69	19.47	37.92
Lead-3	40.34	17.70	36.57

Table 1: ROUGE scores of the best model so far (Celikyilmaz et al., 2018) and Lead-3 baseline on CNN/Daily Mail, non-anonymized version. The Lead-3 baseline is the model that takes the first 3 sentences in the contents as the summary.

Tailoring for this problem, we introduce a new dataset, test the pointer-generated model on this new dataset and further propose two new framework to learn a community-wise abstractive text summarization.

Our paper develops as follows: in Section 2, we will give an overview of the dataset which include our exploratory results. These results also serves

for the motivation of the paper. Section 3 will introduce our base model and the two proposed frameworks. The experiments, results and further discussions are includes in Section 4, 5. The links to the previous literature can be found in Section 6 and we will conclude our findings in Section 7.

2 Data

We use the Webis-TLDR-17 dataset from (Völske et al., 2017). The data is created via the classic notion of "TL;DR" (which stands for "too long, didn't read") in the online discussion website Reddit. For each post consisting "TL;DR", the token "TL;DR", and alike tokens, are naturally considered as separators of the content and the summary.

The data has around 3 million content-summary pairs, together with a few more fields associated with each pair: *'author', 'subreddit', 'title'*, while *'title'* is usually missing. The basic statistics can be found in Table 2.

Subreddits, which are user-created sub forums, covers a huge variety of topics including news, art, science etc. Subreddits are often considered as self-formed clusters of Reddit users. We will also include this information from the filed of *'subreddit'* in our work.

Summarization variation Unlike standard news highlight summaries, the TL;DR summaries exhibit a high level of significant variety. We illustrate this variation with examples shown in Appendix Table 8.

As we see from the examples, the summaries in the data set are generated in more broadly manners. The variation lies in different dimensions among many others:

- Vocabulary usage.
- Abstractness level.
- Key information (attention) distribution.

Community-wise characteristics We believe that the variation mentioned above have community-wise characteristics. Define the 1-gram coverage be the following:

$$G_1(C, S) = \frac{C \cap S}{|S|} \tag{1}$$

where C is the set of content words, S is the set of summary words. In words, $G_1(C, S)$ represents the "abstractness" of the summary with a number between $[0, 1]$, i.e. the lower G_1, the more abstract as the summarization uses new words unseen in the content. For each subreddit, we compute the mean of the 1-gram coverage of the dataset, and plot these mean numbers in a histogram in Figure 1, which shows that there is a nontrivial sign of the the community-wise abstractness difference.[1]

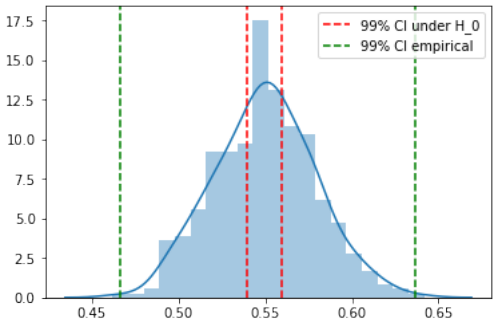

Figure 1: Histogram of the 1-gram cover in each subreddits. The red dash line is the theoretical 99% confidence interval if we assume that there is no community-wise difference. The green dash line is the empirical (true) 99% confidence interval.

Moreover, the variation of abstractness reflects a natural clustering of community. We extract the subreddits with most/least 1-gram cover, and surprisingly, the two clusters share a high in-group similarity and inter-group disparity. See Table 3.

3 Methods

3.1 Baseline Model: Pointer-Generated

We will use the pointer-generated seq2seq model introduced in (See et al., 2017). In this model, a probability p_{gen} is learned at each decoding time, which serves as the weight between using the words from the content and generating new words from the vocabulary.

Note that in the original paper, the authors propose a coverage mechanism aimed at reducing repeating sentences generated by the model. However, (Weber et al., 2018) point out that introducing this coverage mechanism instead will increase the n-gram coverage of the generated summaries and original contents. This is contrary to our desired of learning the abstractness of the new dataset, thus we decide not to include the coverage mechanism, when training and decoding.

[1] In fact, we can show that this difference is statistically significant, assuming that the means are approximately Gaussian.

	min	max	mean	median	std
content length	100	400	212	196	81
summary length	1	395	25	19	23
con_len/sum_len	0.01	0.99	0.13	0.10	0.11

Table 2: Statistics of the content length, summary length and the ratio between them.

Bottom 10			Top 10		
mean G_1	subreddit	category	mean G_1	subreddit	category
0.473	The_Donald	politics	0.629	relationships	relationships
0.483	KotakuInAction	game	0.625	BreakUps	relationships
0.483	ukpolitics	politics	0.621	askwomenadvice	
0.490	PurplePillDebate		0.620	dating_advice	relationships
0.492	europe	politics	0.615	relationship_advice	relationships
0.492	conspiracy	politics	0.614	LucidDreaming	
0.493	FFBraveExvius	game	0.612	Dogtraining	
0.493	TumblrInAction		0.610	LongDistance	relationships

Table 3: Bottom 10 subreddits and top 10 subreddits in mean G_1 score. The subreddits with small G_1 scores are usually controversial with intensive debate (politics-related), or have strong contextual language usage (game-related). The top 10 subreddits are much more benign topics, mostly about relationships or seeking for advice.

3.2 Domain Encoding

As discussed in section 2, we observed a community-wise difference among different subreddits in the TL;DR Dataset. Thus, we would like to incorporate the subreddit information into the training process, which could potentially help us build a community-aware text summarization model with better performance. Our first idea is motivated by Doc2Vec (Le and Mikolov, 2014). Similar to training distributed representation of documents, we train **distributed representation** of communities (i.e. subreddits), which is achieved by augmenting the word representation with the community-specific representation, and then feed the concatenated representation vector into the encoder layer. By doing so, we incorporate the subreddit information as part of the input to the encoder layer.

More specifically, consider the representation for token i of the article is t_i, assume the subreddit information for the article can be represented as a vector $s[j]$, which indicates that this article corresponds to the j-th subreddit. The idea is to concatenate them together and form a longer representation vector. Assume the corresponding weight matrix for the original token $w = [w_1, w_2 \cdots w_n]$ is W_w, we will then create a new learn-able weight matrix W_s for $s = [s_1, \cdots s_{10}]$. The idea is depicted in the **Encoder embedding layer** in Figure 2.

3.3 Multi-Attention

Our second idea is motivated from an empirical study prospective. Even though the attention layer has the fewest number of trainable variables compared to the encoder and decoder layers in the baseline model, it is relatively more important: we find that preforming fine tuning only on the attention layer can help us manage to capture 90% performance improvement compared to preforming the full tuning on the model. Thus we come up with a **multi-attention layer** model training separated attention layers for each subreddit accordingly, which will further lead to an improvement in the overall performance.

The idea is depicted in the **Multi Attention Hidden State** layer in Figure 2.

4 Results

Experiment setting Our model is based on the model by (See et al., 2017), and it is implemented using TensorFlow 1.13. We focus our attention on 10 relatively large and representative subreddit dataset including *"Relationship"*, *"UKPolitics"* and etc. We sample 1200 data point from each of the subreddit and combine them together as our input dataset. We further randomly split the dataset into 80% training set, 10% validation set, and 10% testing set. In addition to our own models, we also

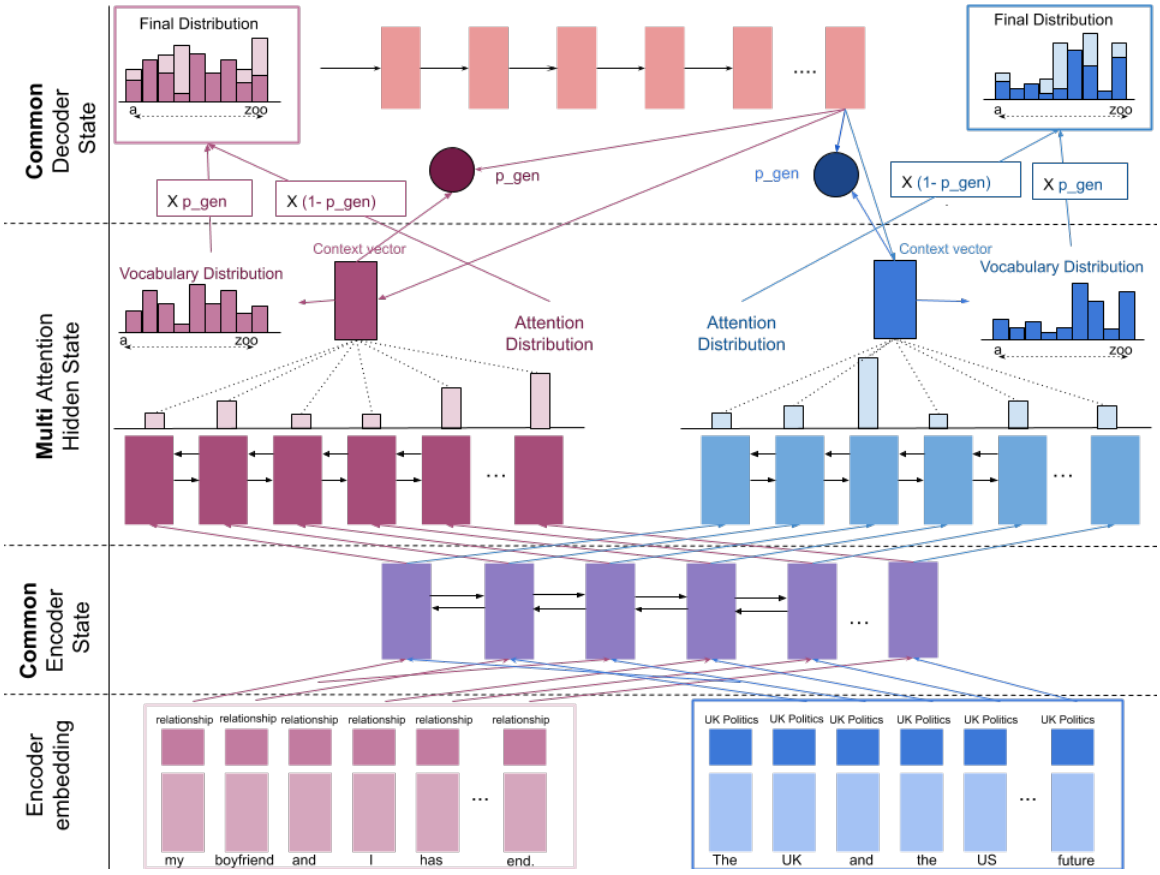

Figure 2: Domain Encoding Multi-Attention Model. Here we consider two sentences from two different subreddits: *Relationship* and *UK Politics*. For each input sentence, we append its subreddit information to the end of each token, and then use the concatenated vectors as the input to the encoder layer; we train multi-attention layers for different subreddits. The vocabulary distribution and the attention distribution are weighted to obtain the final distribution according to the a generation probability $p_{gen} \in [0, 1]$.

report the lead-2 summarization (which uses the two three sentences of the article as a summary) as our baseline result. In the training, all models are initialized using pretrained weights of pointer network. So similarly, our model also uses 256-dimensional hidden states, 128-dimensional word embeddings, and a vocabulary of 50k. We trained the model using Adagrad with learning rate 0.15 and an initial accumulator value of 0.1. We also use gradient clipping with a maximum gradient norm of 2. We use loss on the validation set to implement early stopping.

During training and testing, we truncate the article to 300 tokens and limit the length of the summary to 50 tokens for training and 120 tokens for testing. We trained on a single Tesla K80 and switched to Tesla V100 for pointer network with multi-attention. At test time, summaries are generated using beam search with beam size 4. We trained our model for 10 epochs and it took about 60 minutes on Tesla V100.

Metric We will be using the ROUGE score metric (Lin, 2004). More precisely, ROUGE-n refers to the overlap ratio of n-gram (e.g. when n = 1 it refers to single words) between the system and reference summaries. ROUGE-L refers to the longest common sub-sequence ratio. Specifically, we use a standard evaluation framework for text summarization package called **sumeval**[2] to help us compute the 3 Rouge scores that we mentioned above.

Quantitative performance The ROUGE scores of our models and the baseline models can be found in Table 4. The four models are: lead-2, pointer-generated model with fine tuned on attention, pointer-generated model with Domain Encoding (DE), pointer-generated model with Multi-Attention (MA). As we can see from

[2] https://github.com/chakki-works/sumeval

the results, the fine tuned pointer model and pointer+MA both uniformly beats the lead-2 baseline except form 1 subreddit (*nfl*), while pointer + DC seems to be under-fitted except for 1 subreddit. Between the baseline model and pointer + MA, our model wins on 5 of the subreddits, draws on 3 and lost to the base line model on 1 amongst the other 9 subreddits.

Qualitative performance We provide one specific example from the subrredit *ShouldIbuythisgame* in Table 5. Here we compare the summaries from the baseline model (with full tuning) and our multi-layer attention model. We can see from the original text and the reference that the main subject of this article is to persuade the reader to get the "base game" instead of the dlc. The summary from the baseline model focuses its attention on the first part of the article, resulting in a failure to capture the subject of this article. The summary provided by our model, however, successfully captures the main subject of this article, resulting in a more relavant summary.

5 Discussion

Model size We calculate the proposed two model sizes together with the baseline model in Figure 6. Both models are 30% larger than the baseline model, while the extra parameters are fundamentally different: while the extra attention layers in pointer+MA can be initialized by the pretrained model, the extra weights introduced in pointer-DE have no counterpart in the pretrained model. While the extra degree of freedom introduced in pointer-DE can possibly generate more robust models, it is intrinsically harder to train when the data is insufficient, this is verified in Table 4.

Both models can be improved with smarter use of sizes. For pointer+DE, one can treat the extra dimension as a hyperparameter and tune it with care. For pointer+MA, it is possible that by duplicating **part of** the attention layers can result similar improvement. Due to time constraints, we are not able to finish this line of exploration.

Attention learning To illustrate the logic behind the Multi-attention and further display the community-wise difference, we develop a tool to visualize the *distributional difference of attentions*. Given a *content ratio* $r \in [0, 1]$, let the *precision ratio* $P(r, C, S)$ for the content-summary pair (C, S) be

$$P(r, C, S) = \frac{Precision(C_r, S)}{Precision(C_1, S)} \quad (2)$$

where C_r is the first r (in ratio) part of the content (i.e. C_1 is the whole content). In other words, $P(r, C, S)$ depicts the fraction of the extracted words in the summary from the first r fraction of original text. One can see it as the accumulated density of the total summary precision while scanning through the content. For example, a higher slope near the origin represents the summary concentrate more on the beginning of the content.

We train the baseline model with pretrained-initialization on each subreddit data, with full-tuned (train every weight) and fine-tuned (only train attention layers with same hyperparameters and same epochs. Two examples are shown in Figure 3. By horizontal comparison, we can see that with small amount of data, the fine-tuned model will fit the attention distribution better, i.e. the attention truly capture the attention information; by vertical comparison, we can spot the community difference, i.e. the Reddit users in *ShouldIbuythigame* would put more information in the beginning of their posts. These observations are strong evidence of the validity of our Multi-Attention framework.

Abstraction variation Another dimension we care, as discussed in Section 2, is whether the variation of attractiveness across communities can be learned from our model.

	R-1 std	R-2 std	R-3 std
pointer	3.5	4.8	3.7
pointer + MA	15	16	13

Table 7: Standard deviation of the 1-gram, 2-gram, L-gram precision (using ROUGE-n scores with $\beta = 0.99$) across 10 subreddits.

As we can see from Table 7, our model has one magnitude larger standard deviation of the baseline model.

Technical challenges We also want to list some technical challenges we have faced when building our model. *Memory Limit*: As we increased the model size, the training process requires more GPU memory. As a result, we switched from Tesla

		r	la	nfl	pr	a	S	up	Dt	AH	A
Lead-2	ROUGE-1	15.04	15.86	**15.87**	14.80	16.22	13.88	15.51	15.54	14.27	17.31
	ROUGE-2	2.52	2.93	**2.72**	2.62	3.23	2.61	2.70	2.95	3.19	3.30
	ROUGE-L	10.87	10.80	**11.21**	10.18	11.65	9.83	10.59	10.85	10.15	12.03
pointer fine tuned	ROUGE-1	19.19	**20.42**	14.09	13.89	12.90	**15.68**	15.20	**18.56**	16.90	16.24
	ROUGE-2	5.76	**5.03**	2.60	2.60	2.81	**3.34**	2.80	**4.59**	3.30	3.80
	ROUGE-L	15.40	**15.06**	10.72	10.34	10.38	**12.25**	11.51	**13.52**	12.90	12.79
pointer + DC	ROUGE-1	12.45	9.92	10.76	10.36	**26.16**	12.47	13.25	12.12	13.12	8.11
	ROUGE-2	3.49	1.54	2.10	1.66	**13.86**	2.62	3.15	3.11	2.57	2.12
	ROUGE-L	11.20	8.42	9.46	9.31	**24.51**	10.81	11.44	10.48	11.00	7.49
pointer + MA	ROUGE-1	**21.04**	19.93	14.00	**15.42**	**23.24**	**16.01**	**17.77**	17.78	**18.34**	**17.28**
	ROUGE-2	**6.11**	4.12	2.77	**3.00**	**11.07**	**3.20**	**3.76**	3.76	**3.23**	**4.52**
	ROUGE-L	**16.13**	14.36	10.90	**12.08**	**21.43**	**12.11**	**13.06**	13.05	**13.90**	**13.49**

Table 4: Model performances on 10 subreddits. The subreddits names are shortened. From left to right: *relationships, legaladvice, nfl, pettyrevenge, atheismbot, ShouldIbuythisgame, ukpolitics, Dogtraining, AskHistorians, Anxiety*.

K80 to Tesla V100 for more memory and computation power. We also reduced the batch size, word embedding size, or hidden layer dimension in LSTM in the experiments.

Time Limit: Due to the computation resource, we limit the size of dataset so that we can finish the training within reasonable time. But we expect to see better results with larger dataset.

6 Related Work

Text summarization Beyond our baseline model (See et al., 2017), various attempts have been made to build better text summarization models. Improvements are focused on the encoder/decoder (Jadhav and Rajan, 2018; Celikyilmaz et al., 2018), attention mechanism (Nallapati et al., 2017; Al-Sabahi et al., 2018; Li et al., 2018a; Gehrmann et al., 2018; Li et al., 2018b) and incorporating external knowledge (Guo et al., 2018; Cao et al., 2018; Kryściński et al., 2018). Another important technical development is introducing reinforcement learning when training the model, which is first implemented in (Paulus et al., 2017; Rennie et al., 2017). By designing task-specific objective functions, the idea of using RL training is often coupled with solving other tasks alongside with improving ROUGE scores, for example, coherence (Wu and Hu, 2018; Celikyilmaz et al., 2018), saliency and entailment (Pasunuru and Bansal, 2018; Chen and Bansal, 2018) and abstraction (Kryściński et al., 2018) among many others.

Domain adaptation Our work is also related to the problem of domain adaptation, which essentially studies how to transfer the knowledge learn from one dataset to another. Our idea of using pre-trained model to initialize the encoder and decoder comes from the work (Ramachandran et al., 2016), in which the authors use a pre-trained language model as the encoder/decoder initialization and prove that it will boost the performance of the seq2seq text summarization model, also see (Radford et al., 2018). On transferring the information in the learned attention mechanism, various attempts are made in multiple tasks and communities, for example, sentiment analysis (Li et al., 2018c) and computer vision (Zagoruyko and Komodakis, 2016; Wang et al., 2019).

Task-specific ensemble We also want to mention the correlation to the idea of aggregating or learning from task/domain-specific information. It motivate huge success in problems like multi-task visual tracking (Nam and Han, 2016), in which the authors design a networking consisting task-shared layers and task-separated layers, which also motivates our work. Ensemble of multi-domain information is also studied in NLP, for example (Kim et al., 2017, 2018) tailoring for speech problems. In text summarization, the closest work to ours is (Celikyilmaz et al., 2018), where the authors design a muli-agent network in the encoder layers with each agent targeting on part of the content paragraphs and being aggregated with a agent-attention.

Category	Content
Original Text	the dlc is all just a waste of money imo . it 's just outfits or weapons or stuff for the multi-player . there 's one 5 minute long sp map , which is n't worth it for the price . the base game is fantastic however , i loved it when i played it on my laptop last year during a lull in my finals and i revisited it yesterday now that i 've built a proper rig . i intended to just run around a bit and admire the visuals with tressfx and the like , but i ended up getting lost in the game for longer than i intended . it 's seriously great and i ca n't recommend it enough .
Baseline Summary	the dlc is just a waste of money . it 's just a outfits or weapons or stuff for the multi-player . it 's seriously great and i ca n't recommend it enough . it 's seriously great and i ca n't recommend it enough .
Our model Summary	base game is seriously great and i ca n't recommend it . it 's worth it . it 's worth it for the multi-player . it 's worth it . it 's worth it .
Reference	get the base game .

Table 5: One examples of an article in the subrredit *ShouldIbuythisgame*. Two summaries from the baseline model and our model are provided. The reference is provided by the author as the truth summarization. Our model successfully captures the main subject of this article which is the base game.

	encoder	attention	decoder
pointer	7.4m	0.78m	13m
pointer + DE	13.8m	0.78m	13m
pointer + MA	7.4m	7.8m	13m

Table 6: Model size decomposed by model components. The sizes are computed with the default hyper-parameters setting.

7 Conclusion

In this work, we present a community specific pointer generator model, and show that it reduces inaccuracy in terms of which part of the content it should pay attention to. We applied our model to a new and challenging TL;DR Reddit dataset, which are more lingual diverse and exhibit community-wise difference among each subreddit, and significantly outperforms the single attention layer abstractive state-of-art result. Our model exhibits abilities of focusing its attention differently for different subreddit sets, resulting an overall improvement in terms of the general performance.

Possible future works includes: 1. combining the Community Encoding and Multi-Attention into the same model; 2. improving the current framework by further more detailed layer selection as we discussed in Section 5; 3. develop better way to encode the community information, e.g. using the techniques including Doc2Vec and better word embeddings.

Authorship Statement

Yiyang writes most of the codes. Teng develops the main model and the statistical exploration. Yatong joins the brainstorming, debugging, data pre-processing and our final report. We equally contribute to the work.

References

Kamal Al-Sabahi, Zhang Zuping, and Mohammed Nadher. 2018. A hierarchical structured self-attentive model for extractive document summarization (hssas). *IEEE Access*, 6:24205–24212.

Ziqiang Cao, Wenjie Li, Sujian Li, and Furu Wei. 2018. Retrieve, rerank and rewrite: Soft template based neural summarization. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 152–161.

Asli Celikyilmaz, Antoine Bosselut, Xiaodong He, and Yejin Choi. 2018. Deep communicating agents

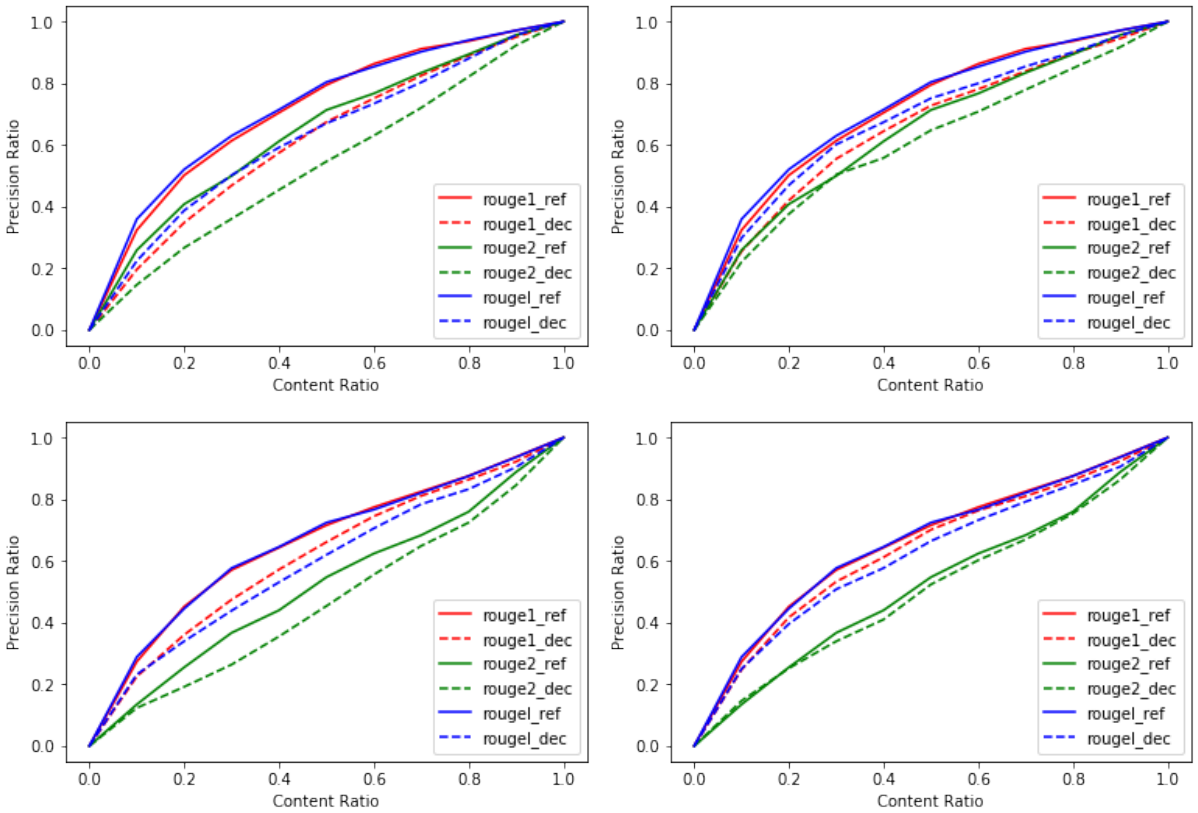

Figure 3: Precision ratio vs content ratio plots. From top to down, left to right: full-tuned in *ShouldIbuythisgame*, fine-tuned in *ShouldIbuythisgame*, full-tuned in *Anxiety*, fine-tuned in *Anxiety*. The fine-tuned models can learn the distribution better.

for abstractive summarization. *arXiv preprint arXiv:1803.10357*.

Yen-Chun Chen and Mohit Bansal. 2018. Fast abstractive summarization with reinforce-selected sentence rewriting. *arXiv preprint arXiv:1805.11080*.

Sebastian Gehrmann, Yuntian Deng, and Alexander M Rush. 2018. Bottom-up abstractive summarization. *arXiv preprint arXiv:1808.10792*.

Max Grusky, Mor Naaman, and Yoav Artzi. 2018. Newsroom: A dataset of 1.3 million summaries with diverse extractive strategies. *arXiv preprint arXiv:1804.11283*.

Han Guo, Ramakanth Pasunuru, and Mohit Bansal. 2018. Soft layer-specific multi-task summarization with entailment and question generation. *arXiv preprint arXiv:1805.11004*.

Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *Advances in neural information processing systems*, pages 1693–1701.

Aishwarya Jadhav and Vaibhav Rajan. 2018. Extractive summarization with swap-net: Sentences and words from alternating pointer networks. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 142–151.

Young-Bum Kim, Dongchan Kim, Anjishnu Kumar, and Ruhi Sarikaya. 2018. Efficient large-scale neural domain classification with personalized attention. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2214–2224.

Young-Bum Kim, Karl Stratos, and Dongchan Kim. 2017. Domain attention with an ensemble of experts. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 643–653.

Wojciech Kryściński, Romain Paulus, Caiming Xiong, and Richard Socher. 2018. Improving abstraction in text summarization. *arXiv preprint arXiv:1808.07913*.

Quoc Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *International conference on machine learning*, pages 1188–1196.

Chenliang Li, Weiran Xu, Si Li, and Sheng Gao. 2018a. Guiding generation for abstractive text summarization based on key information guide network. In

Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers), pages 55–60.

Wei Li, Xinyan Xiao, Yajuan Lyu, and Yuanzhuo Wang. 2018b. Improving neural abstractive document summarization with explicit information selection modeling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1787–1796.

Zheng Li, Ying Wei, Yu Zhang, and Qiang Yang. 2018c. Hierarchical attention transfer network for cross-domain sentiment classification. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. *Text Summarization Branches Out*.

Ramesh Nallapati, Feifei Zhai, and Bowen Zhou. 2017. Summarunner: A recurrent neural network based sequence model for extractive summarization of documents. In *Thirty-First AAAI Conference on Artificial Intelligence*.

Hyeonseob Nam and Bohyung Han. 2016. Learning multi-domain convolutional neural networks for visual tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4293–4302.

Ramakanth Pasunuru and Mohit Bansal. 2018. Multi-reward reinforced summarization with saliency and entailment. *arXiv preprint arXiv:1804.06451*.

Romain Paulus, Caiming Xiong, and Richard Socher. 2017. A deep reinforced model for abstractive summarization. *arXiv preprint arXiv:1705.04304*.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training. *URL https://s3-us-west-2. amazonaws. com/openai-assets/research-covers/languageunsupervised/language understanding paper. pdf*.

Prajit Ramachandran, Peter J Liu, and Quoc V Le. 2016. Unsupervised pretraining for sequence to sequence learning. *arXiv preprint arXiv:1611.02683*.

Steven J Rennie, Etienne Marcheret, Youssef Mroueh, Jerret Ross, and Vaibhava Goel. 2017. Self-critical sequence training for image captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7008–7024.

Abigail See, Peter J Liu, and Christopher D Manning. 2017. Get to the point: Summarization with pointer-generator networks. *arXiv preprint arXiv:1704.04368*.

Michael Völske, Martin Potthast, Shahbaz Syed, and Benno Stein. 2017. Tl; dr: Mining reddit to learn automatic summarization. In *Proceedings of the Workshop on New Frontiers in Summarization*, pages 59–63.

Ximei Wang, Liang Li, Weirui Ye, Mingsheng Long, and Jianmin Wang. 2019. Transferable attention for domain adaptation.

Noah Weber, Leena Shekhar, Niranjan Balasubramanian, and Kyunghyun Cho. 2018. Controlling decoding for more abstractive summaries with copy-based networks. *arXiv preprint arXiv:1803.07038*.

Yuxiang Wu and Baotian Hu. 2018. Learning to extract coherent summary via deep reinforcement learning. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Sergey Zagoruyko and Nikos Komodakis. 2016. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. *arXiv preprint arXiv:1612.03928*.

A Appendices

Example	Extractive summary	Abstractive summary	Variation
Content	It doesn't sound like you have a realistic view of the economy. The world's wealth is not a zero-sum game. It's the same reason that people can invest in the stock market long term and ALWAYS expect growth (over years, possibly decades). The economy grows. ... Believe it or not, this is not a bad time to be poor at all. Google it if you don't believe me. Lots of sources say that this century is the most peaceful, just to give another comparison.	I used to work for MGM Studios (now known as Disney's Hollywood Stuidos,) and it was Soap Weekend. ... I was heading into the castmember restroom after a delicious lunch of mystery nuggets. The door seemed a bit stuck, so I slammed into it like a line-backer on steroids. Unbeknownst to me, Susan Lucci was on the other side, trying to escape our filth-laden bathroom. My beefy body hit the door, and I heard an "oof" as yards of skanky red cloth, shiny brunette hair and about 17 inches of spike heels flew across the room and landed in the sink.	To simply say that The Massive Monkees were better dancers is a bit of an overstatement. Hitting the moves on the beat is the basics of not only breakdancing, but all dancing. The Massive Monkees didn't just beat some group of guys that were good at stunts, they beat THE Jinjo Crew. ... It gets easy for things to get biased when we put thing in terms of "USA vs Korea" or Country X vs Country Y. ... They're excited for each other and freaking hugging each other and shit. We have to remember, or at least try to remember, that's what this is about. Bringing the world together by sharing in what we love to do.
Summary	World GDP is not a zero-sum game. This is a great time to be poor and in "despair."	I knocked Susan Lucci on her ass.	Oppan Gangnam Style
Subreddit	YouShouldKnow	AskReddit	videos

Table 8: 3 examples of summarization variability. Highlighted texts are similar texts in the content and summary. The size of the contents are tailors to fit in the table. Example 1 is the "classic", extrative summarization, and the key sentences appear in the beginning of the paragraphs in the content. 2 is an example of abstractive summarization, which summarizes the experience with Susan Lucci with the words "knock on her ass". The summary in 3 has zero overlap with the content. Some contextual knowledge will inform us that the author of the post use this lyric as a backup of his "bring the world together" claim since the dancer group Massive Monkees played Korean-style show to win the American dance competition.